

# Máster Avanzado de Programación en Python para Hacking, BigData y Machine Learning

Desarrollo y gestión ágil de proyectos Python

## LECCIÓN 02

# Desarrollo de proyectos ágiles con Scrum

# Gestión ágil

La **gestión ágil** surgió como antítesis a un modelo de gestión tradicional: la gestión de proyectos predictiva.

Tanto la gestión ágil, como la predictiva o tradicional tienen sus virtudes y resultan más útiles en ciertas industrias.

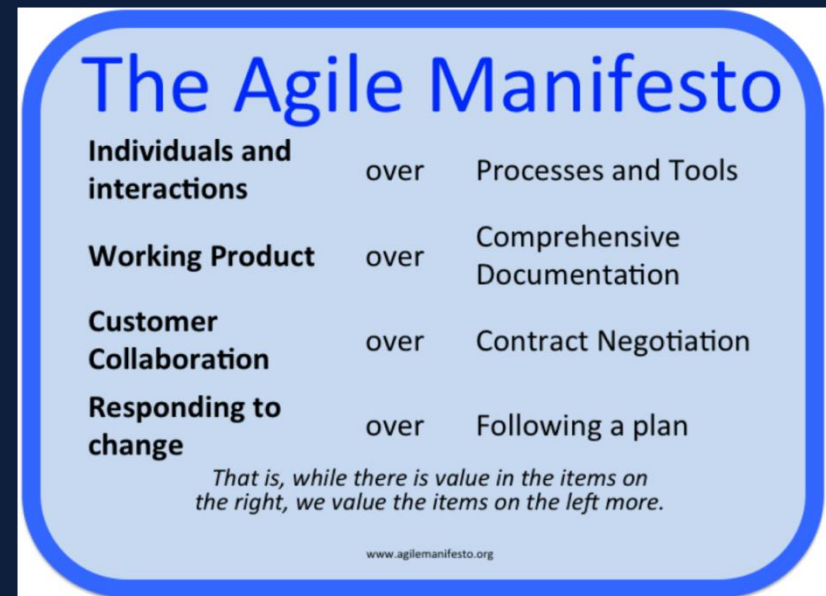
La gestión predictiva se centra en planificar, en calcular un presupuesto y marcar plazos de entrega. Si el proyecto final se termina en la fecha acordada, sin exceder el coste y con todas las funcionalidades del plan inicial, se considera un éxito.

Por muy razonable que suene esta estrategia, si trabajamos en industrias que se caractericen por su constante y rápida evolución (como es el desarrollo de software), encontraremos que tiene muchos inconvenientes. Esa definición de un proyecto exitoso sirve en un entorno estable, con productos que son resultado de una atención escrupulosa a procesos y protocolos.

## El manifiesto ágil

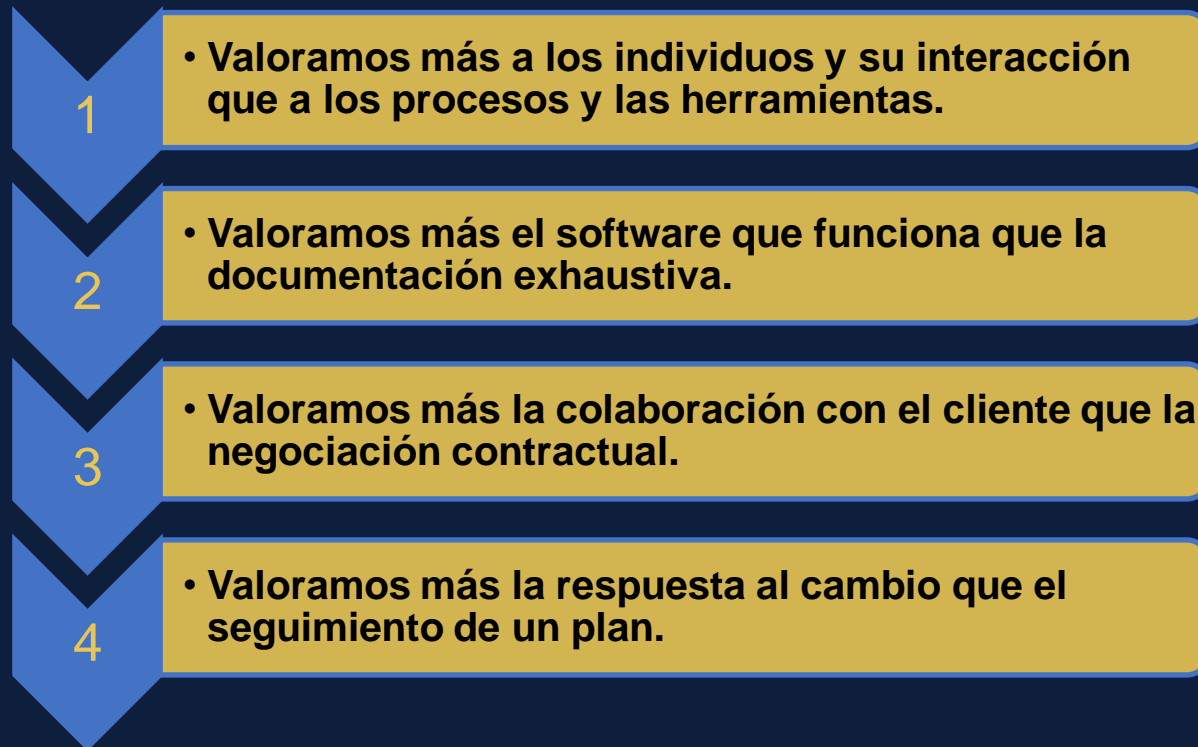
El **manifiesto ágil** es un documento que fue redactado y publicado en 2001 por 17 personas destacadas en la industria del software que se reunieron para debatir sobre las técnicas y procesos empleados por los equipos de programación para desarrollar software, este documento se compone de cuatro valores fundamentales y doce principios que lideran el enfoque ágil para el desarrollo de software.

<https://agilemanifesto.org/iso/es/manifesto.html>



## El manifiesto ágil

4 postulados (valores)



## El manifiesto ágil

### 12 principios

#### 12 AGILE PRINCIPLES

- |   |   |   |
|---|---|---|
| <b>01</b> Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. | <b>02</b> Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. | <b>03</b> Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.                |
| <b>04</b> Business people and developers must work together daily throughout the project.                             | <b>05</b> Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. | <b>06</b> Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. |
| <b>07</b> Working software is the primary measure of progress.  | <b>08</b> The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.   | <b>09</b> Continuous attention to technical excellence and good design enhances agility.  |
| <b>10</b> Simplicity – the art of maximizing the amount of work not done – is essential.                              | <b>11</b> The best architectures, requirements, and designs emerge from self-organizing teams.  | <b>12</b> At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.                     |

# El manifiesto ágil

1. Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
2. Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se dobligan al cambio como ventaja competitiva para el cliente.
3. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.
4. Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
5. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software que funciona es la principal medida del progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica enaltece la agilidad.
10. La simplicidad como arte de maximizar la cantidad de trabajo que se hace, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se autoorganizan.
12. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia.

# Scrum

Scrum es un framework para desarrollar proyectos ágiles de manera iterativa e incremental.

El nombre “scrum” viene de una jugada de Rubby.

Al igual que en el juego, el equipo Scrum es parte fundamental del trabajo y no debe nunca derrumbarse ni romperse.





# Scrum

Scrum proporciona calidad, rapidez en la entrega y bajos costes, evita la burocracia y la documentación, de manera que los primeros resultados lleguen muy rápidamente.

*¿Y cómo se consigue esto?* Mediante la experiencia.

Scrum se basa en la teoría de control de procesos empírica o **empirismo**.

El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce.

Scrum emplea un enfoque **iterativo e incremental** para optimizar la predictibilidad y el control del riesgo.

Tres pilares soportan toda la implementación del control de procesos empírico:

1. Transparencia
2. Inspección
3. Adaptación

## Scrum

Scrum es un marco de trabajo que se basa en la mejora continua y es, precisamente, por ello por lo que aún manteniendo su esencia original evoluciona continuamente.

Equipo scrum, compuesto de los siguientes **roles**:

- ☐ Equipo de desarrollo (desarrolladores y testers).
- ☐ Propietario del producto (product owner).
- ☐ Scrum master.
- ☐ Interesados (stakeholders).

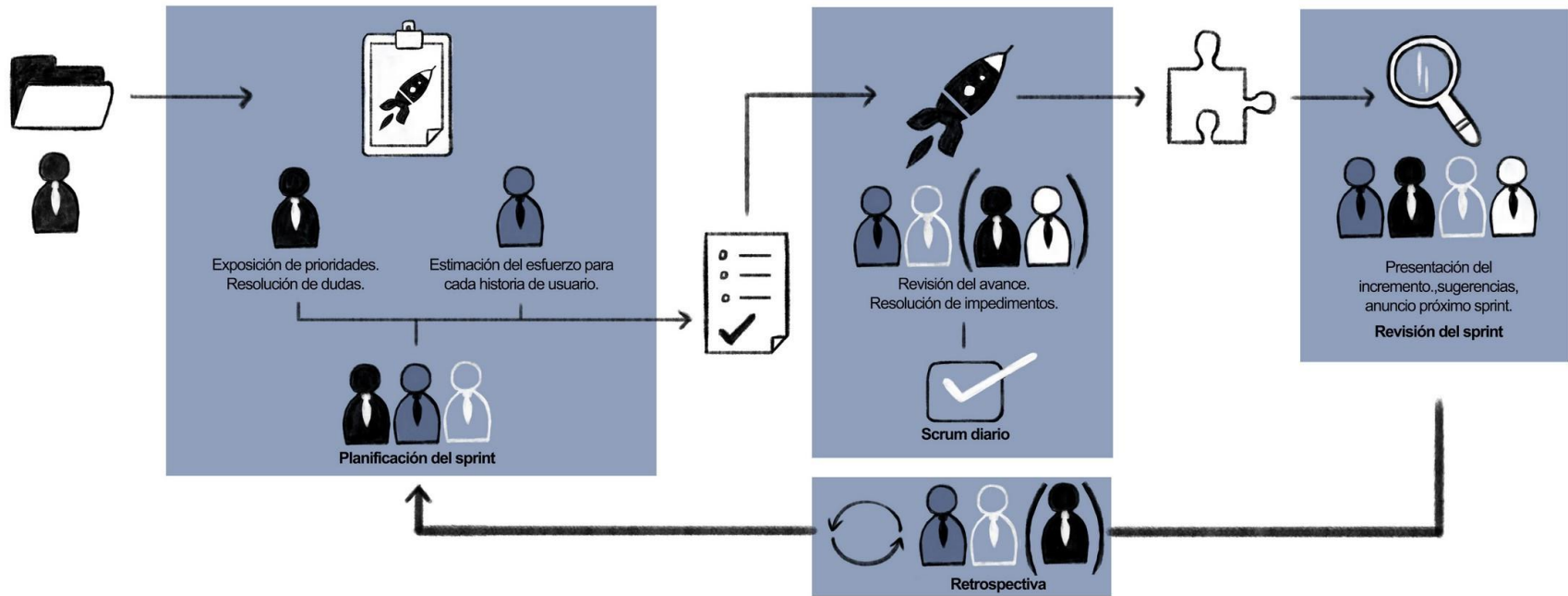
**Artefactos:**

- ☐ Pila del producto (product backlog).
- ☐ Pila del sprint (sprint backlog).
- ☐ Incremento.

**Eventos:**

- ☐ Sprint.
- ☐ Reunión de planificación del sprint (sprint planning).
- ☐ Scrum diario (sprint daily o stand up).
- ☐ Revisión del sprint (sprint review)
- ☐ Retrospectiva del sprint (sprint retrospective).

## LAS REGLAS DE SCRUM



### ROLES

- PROPIETARIO DEL PRODUCTO**  
Determina las prioridades.  
Una sola persona.
- DESARROLLADOR**  
Construye el producto.
- SCRUM MASTER**  
Gestiona y facilita la ejecución de las reglas del Scrum.
- INTERESADOS**  
Resto de implicados. Asesoran y observan.

### ARTEFACTOS

- PILA DEL PRODUCTO**  
Relación de requisitos del producto, no es necesario excesivo detalle. Priorizados. Lista en evolución y abierta a todos los roles. El propietario del producto es su responsable y quien decide.
- PILA DEL SPRINT**  
Requisitos comprometidos por el equipo para el sprint con nivel de detalle suficiente para su ejecución.
- INCREMENTO**  
Parte del producto desarrollada en un sprint, en condiciones de ser usada (pruebas, codificación limpia y documentada).

### EVENTOS

- PLANIFICACIÓN DEL SPRINT**  
1 jornada de trabajo (máx.) El propietario del producto explica las prioridades. El equipo estima el esfuerzo de los requisitos prioritarios y se elabora la pila del sprint. El equipo define en una frase el objetivo del sprint.
- SPRINT**  
Ciclo de desarrollo básico en el marco estándar de scrum, de duración recomendada inferior a un mes y nunca mayor de 6 semanas.
- SCRUM DIARIO**  
15 minutos máximo. Responsabilidad del equipo. cada miembro expone: Lo que hizo ayer, lo que va a hacer hoy y si tiene o prevé problemas. Se actualiza la pila del sprint.
- REVISIÓN DEL SPRINT**  
Informativa, máximo 4 horas. Presentación del incremento, planteamiento de sugerencias y anuncio del próximo sprint.
- RETROSPECTIVA**  
El equipo autoanaliza la forma de trabajo. Identificación de fortalezas y debilidades. Refuerzo de las primeras, plan de mejora de las segundas.

# Roles de Scrum

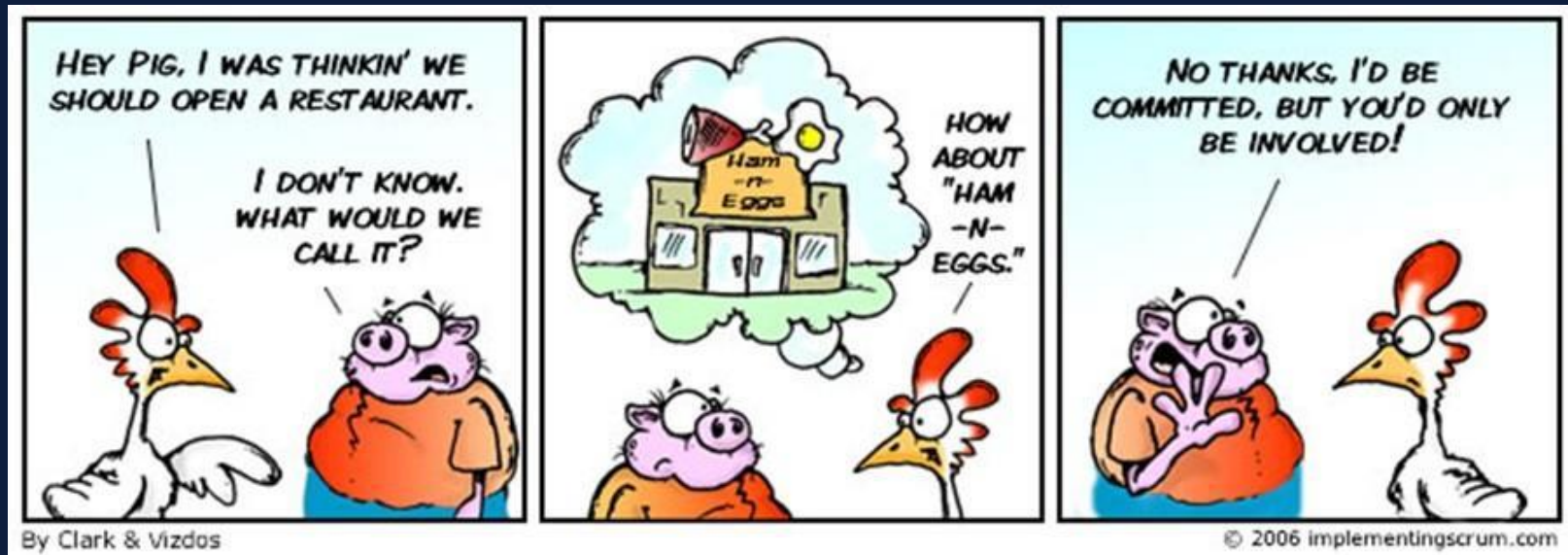
Los roles en un marco de scrum técnico son:

- Propietario del producto
- Desarrollador
- Scrum Master

Todas las personas que intervienen, o tienen relación directa o indirecta con el proyecto, se clasifican en dos grupos: comprometidos e implicados.

En círculos de Scrum es frecuente llamar a los primeros (sin la menor connotación peyorativa) “cerdos” y a los segundos “gallinas”. El origen de estos nombres está en la metáfora que ilustra de forma gráfica la diferencia entre “compromiso” e “implicación” con el proyecto.

## Roles de Scrum



*Una gallina y un cerdo paseaban por la carretera. La gallina preguntó al cerdo: “¿Quieres abrir un restaurante conmigo?”. El cerdo consideró la propuesta y respondió: “Sí, me gustaría. ¿Y cómo lo llamaríamos?”. La gallina respondió: “Jamón con huevos”. El cerdo se detuvo, hizo una pausa y contestó: “Pensándolo mejor, creo que no voy a abrir un restaurante contigo. Yo estaría realmente comprometido, mientras que tu estarías sólo implicada”.*

## Roles de Scrum

COMPROMETIDOS (CERDOS)	IMPLICADOS (GALLINAS)
Propietario del producto	Otros interesados (dirección, gerencias, comerciales, marketing, etc.)
Miembros del equipo	

Una observación en este punto, sobre el rol de Scrum Master, por ser en ocasiones frecuente la duda de considerar si es un rol “comprometido” o “implicado”.

Partiendo de que la división entre personas comprometidas y personas implicadas es más “conceptual” que “relevante”, pero cuando se trabaja con este rol presente, su responsabilidad es el funcionamiento de un scrum técnico en la organización.

Su responsabilidad directa, su misión, es por tanto la forma de trabajo, siendo por tanto el producto elaborado en los proyectos un objetivo de segundo nivel, o indirecto.

Por esta razón en el cuadro anterior no se considera el rol de Scrum Master, aunque que en cualquier caso no es una cuestión especialmente relevante. Si hubiera que forzar una respuesta, desde el criterio de que no está comprometido en el proyecto (sino en la mejora de la forma de trabajo) se debería considerar como un rol “implicado” .

Sin embargo, en la práctica, el Scrum Master es una figura muy implicada en el equipo y el proyecto en sí mismo; por tanto, sería “comprometido”.

## Roles de Scrum

How people reacts differently to a single word.

**"Bug"**



Tester



Developer



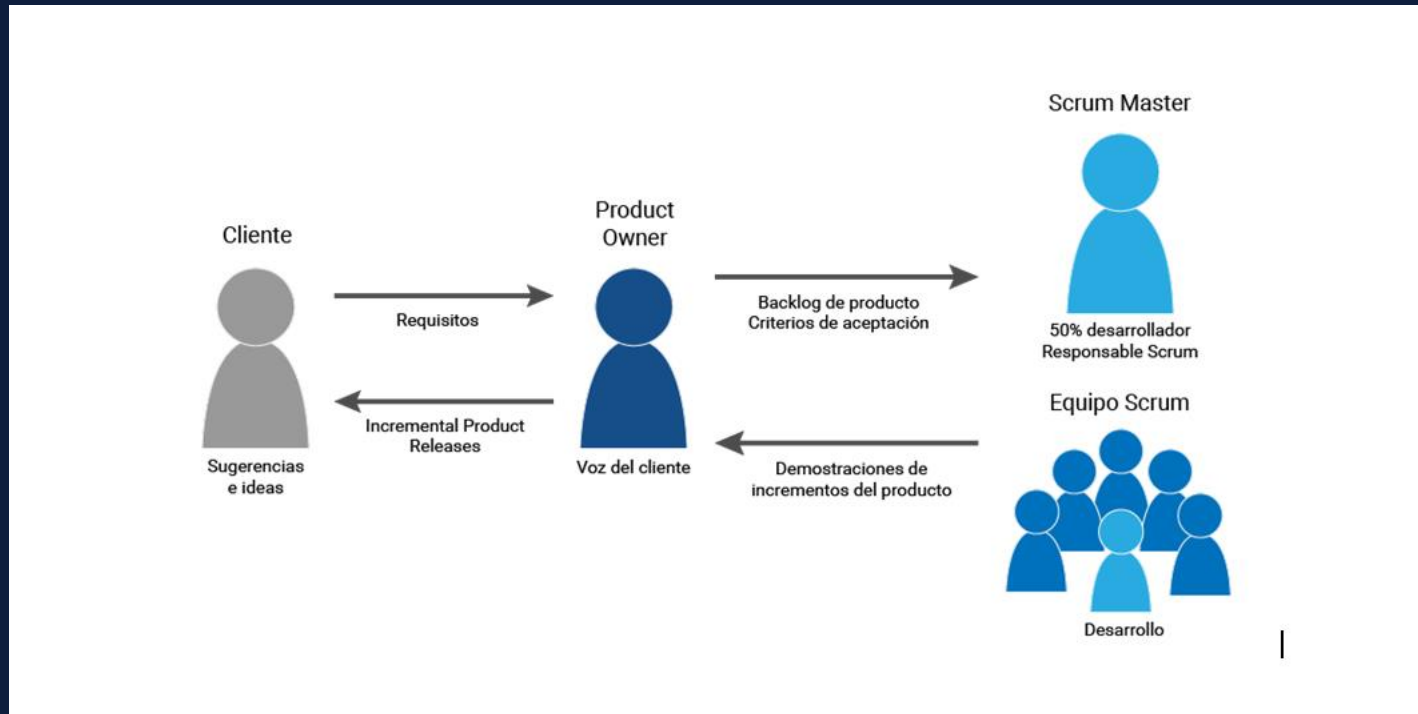
Manager

How people react differently to the word "bug"





## Roles de Scrum





MUCHAS GRACIAS POR SU ATENCIÓN



mjpena@grupomainjobs.com



María José Peña Carrilero

<https://www.linkedin.com/in/maria-jose-pe%C3%B1a-carrilero/>



[twitter.com/eiposgrados](https://twitter.com/eiposgrados)



[facebook.com/eiposgrados](https://facebook.com/eiposgrados)



[instagram.com/eiposgrados](https://instagram.com/eiposgrados)